REMARKS/ARGUMENTS

Status of the Claims

- Claims 1-87 are pending in the Application.
- Claims 1-87 are rejected by Examiner.

Claim Rejections Pursuant to 35 U.S.C. §103(a)

Claims 1-7, 12-14, 16-30, 35-37, 39-46, 51-53, 55-62, 67-69, 71-78, 83-85, and 87 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Brumme (U.S. Patent No. 6,134,559) in view of Hejlsberg (U.S. Patent No. 6,185,728). It is respectfully submitted that the claims are patentable for the reasons set forth below.

Independent claim 1 teaches:

> A method for operating a computer using object-based computer code, the method comprising:
> *invoking an event handler method, using a delegate*, by calling another method of an instance of a class for which parameters passed to the other method are also passed to the event handler method, a parameter list of the other method having a same signature as a parameter list of the event handler method, wherein the other method references the event handler method, and the delegate contains a reference to the other method;
> creating an invocation list associated with the other method, the invocation list specifying one or more event handler methods to be invoked; and
> dynamically altering contents of the invocation list. (emphasis added)

The present invention, as embodied in claim 1, uses a delegate-based event driven programming model to handle events (Application as filed, page 10, ll. 3-4). A delegate is an object that contains a pointer to a method, as well as a pointer to an object that the referenced method is to be applied (Id., page 10, ll. 5-7). None of the prior art, taken alone or in combination, discloses or suggests such features.

Brumme describes a system and method for integrating objects of foreign type systems into a single unified type system (Brumme, col. 4, ll. 2-4). Foreign object adapters are configured to support one or more foreign object systems (Id., col. 4, ll. 19-21). The foreign object adapters convert a class type from the foreign object system to a class type compatible with the single unified type system (Id., col. 4, ll. 39-41).

Hejlsberg teaches a visual development system which allows a developer/user to easily control object behavior, whether the user is working in a visual environment or a programming environment, or switching back and forth between the two (Hejlsberg, col. 3, ll. 43-47). The system features method pointers that allow a developer or programmer to achieve delegation between objects programmatically as well as visually (Id., col. 3, ll. 47-51).

The Examiner admits that Brumme does not teach using a delegate (Office Action, page 3). The Examiner states Hejlsberg teaches the step of delegation of an event handler (Office Action, page 3). To the contrary, it is respectfully submitted that while Hejlsberg teaches delegation of an event handler using method pointers, where a method pointer is assigned, for a particular event of an object to point to a particular handler, it does not teach the use of a delegate as required by the claims (Hejlsberg, col. 8, ll. 1-10). In the present invention, the delegation is achieved using a specific delegate object containing a pointer to a method, and a pointer to an object to which the method is to be applied (Application as filed, page 4, ll. 15-18).

In addition, even if Hejlsberg taught using a delegate object, there is no motivation to combine the system of Brumme with the system of Hejlsberg. Brumme is directed towards a uniform object model that integrates objects defined by foreign type systems into a single integrated object oriented system (Brumme, col. 4, ll. 1-5). Hejlsberg is directed towards a visual development system where the user is able to switch between visual and programming environments (Hejlsberg, col. 3, ll. 42-51). Since Brumme is directed towards a system where objects of foreign types can be integrated into a single type system, there would have been no motivation to combine the system with Hejlsberg, which is directed to systems and methods where users can easily switch been programming in a visual environment and a more traditional environment, because the systems are directed towards completely different purposes. Applicant respectfully requests that the Examiner withdraw the 35 U.S.C. § 103(a) rejection and allow claim 1.

Independent claims 17, 24, 40, 56, and 72 contain similar limitations to claim 1 and are therefore allowable for the same reasons given for claim 1. It is requested that the Examiner withdraw the rejections and allow claims 17, 24, 40, 56, and 72.

With respect to rejected dependent claims 2-7, 12-14, 16, 18-23, 25-30, 35-37, 39, 41-46, 51-53, 55, 57-62, 67-69, 71, 73-78, 83-85, and 87, they are all variously dependent on independent claims 1, 17, 24, 40, 56, 72 and are therefore all allowable for at least the reasons given above. Applicant respectfully requests that the Examiner withdraw the 35 U.S.C. § 103(a) rejections and allow the claims.

Claims 8-11, 31-34, 47-49, 50, 63-66, 79-82 stand rejected under 35 U.S.C § 103(a) as being unpatentable over Brumme in view of Hejlsberg and in further view of Wold (U.S. Patent No. 5,724,589). It is respectfully submitted that these claims are patentable for reasons set forth below.

Wold describes a system for providing a property-method-event programming model for developing context-free reusable software components (Wold, col. 3, ll. 57-60). The system allows developers to create pre-packaged C++ software components that can be plugged into an existing design (Id., col. 3, ll. 61-64).

As described above, Neither Brumme nor Hejlsberg, alone or in combination, teach invoking an event handler using a delegate as required by the claims. Wold fails to cure the deficiencies of Brumme and Hejlsberg. It is therefore respectfully requested that the Examiner withdraw the rejection and allow claims 8-11, 31-34, 47-49, 50, 63-66, and 79-82.

Claims 15, 38, 54, 70, and 86 stand rejected under 35 U.S.C § 103(a) as being unpatentable over Brumme in view of Hejlsberg and in further view of Kimura (U.S. Patent No. 6,292,849). It is respectfully submitted that these claims are patentable for reasons set forth below.

Kimura describes a system for sharing objects by multiple application programs (Kimura, col. 2, ll. 13-15). The system comprises plural control objects used as common objects, each object further comprising a first function for serving properties to an application program, another control object, and a second function for posting events (Id., col. 2, ll. 15-20).

As described above, Neither Brumme nor Hejlsberg, alone or in combination, teach invoking an event handler using a delegate as required by the claims. Kimura fails to cure the

deficiencies of Brumme and Hejlsberg. It is therefore respectfully requested that the Examiner withdraw the rejection and allow claims 15, 38, 54, 70, and 86.

CONCLUSION

In view of the above remarks, Applicants respectfully submit that the present application is in condition for allowance. Reconsideration of the application and an early Notice of Allowance are respectfully requested.

Date: August 11, 2004

Jonathan M. Waldman
Registration No. 40,861

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439